

VECTOR PERFORMANCE ANALYSIS OF THE NEC SX-2

Rod A. Fatoohi¹

Report RNR-89-013, October 1989

*Sterling Software
NASA Ames Research Center
Moffett Field, California 94035*

Abstract. This paper presents the results of a series of experiments to study the vector performance of the NEC SX-2. The main object of this study is to understand the architecture and identify its bottlenecks and limiting factors. A simple performance model is used to examine the impact of certain architectural features on the performance of a set of basic operations. The results of implementing this set on the machine for four vector lengths and three memory strides are presented and compared. These results show that the vector length and the ratio of floating point operations to memory references have a great impact on the performance of the machine. Two numerical algorithms are also employed and the results of these algorithms and the basic operations are compared to early results on one processor of the Cray-2 and Cray Y-MP. These comparisons show that the SX-2 is faster than the Cray Y-MP by up to 86% for short vectors and by 2 to 4 times for long vectors. Also, it outperformed the Cray-2 by even bigger factors. Finally, the architecture of the SX-X is presented, and some predictions about its performance are given.

¹ This research was supported by NASA Contract No. NAS2-11555 while the author was an employee of Sterling Software under contract to the NAS Applied Research Office at NASA Ames Research Center.

1. Introduction

There has been a lot of interest in the performance analysis of supercomputers. The performance of the Japanese supercomputers is no exception. Among the recent studies is the work by Lubeck et al. [5] to compare the performance of the Fujitsu VP-200 and Hitachi S810/20 with the Cray X-MP/2 using the Los Alamos benchmark set. In a follow-up article, Lubeck et al. [4] studied the performance of the NEC SX-2 using the same benchmark set. Their study showed the SX-2 is 1.5 to 3 times faster on short vectors and 2 to 4 times faster on long vectors than one processor of the Cray X-MP with a 9.5 nsec clock period. Van Kats et al. [6] compared the performance of the SX-2 with the Hitachi S810/20, Fujitsu VP-200, and one processor of the 9.5 nsec Cray X-MP using simple operations, the Livermore kernels, and some linear algebra codes. Their results showed that the SX-2 is consistently 2 to 2.5 times faster than the other machines. The results of running NAS Kernels [1] showed that the SX-2 is about twice as fast as one processor of the Cray Y-MP and about 2.5 times faster than one processor of the 8.5 nsec Cray X-MP.

This paper is a follow-up to the study of the vector performance analysis of the Cray-2, Cray Y-MP, and ETA10-Q [2]. Both studies are focused on the impact of certain architectural features on the performance of supercomputers. The same set of experiments were conducted on these machines. In this paper, the emphasis is on the SX-2 and its performance relative to the Cray Y-MP and Cray-2. The machine architecture is briefly described in section 2. The results of implementing a set of basic operations for four vector lengths and three memory strides are presented in section 3. Section 4 presents a simple performance model to determine

the impact of certain architectural features on the performance of the machine. Section 5 presents a comparison of performance of the SX-2 with one processor of the Cray-2 and Cray Y-MP. The results of implementing two numerical algorithms are presented and compared in section 6. Section 7 presents the architecture of the new NEC machine, the SX-X, and some predictions about its performance. Finally section 8 contains some concluding remarks.

2. The NEC SX-2

The NEC SX-2 is a register-to-register vector supercomputer with an arithmetic processor, a control processor, an I/O processor, a main memory, and an extended memory [7]. The arithmetic processor has separate scalar and vector units to execute user codes. The control processor handles job scheduling and resource management, and works independently of the arithmetic processor. The I/O processor can connect up to 32 channels. The main memory has up to 128 Mwords (64-bit) organized in 512 banks. It uses 256 kbit SRAM chips with a 40 nsec access time. An extended memory of up to 1024 Mwords is also available with a peak transfer rate of 1.3 Gbytes/s. The basic clock period (CP) of the machine is 6 nsec. This work was performed using the machine at the Houston Area Research Center which runs the SXOS operating system. The vectorizing compiler is Fortran77/SX.

The vector unit has some interesting features. It has four identical sets of functional units. Each set computes every forth element of the same vector operation. Each set has four pipelined units: add, multiply, logical, and shift units. These functional units receive their operands from and deliver their results to 40

vector registers of 256 elements each. There are two paths between the vector registers and main memory: a load path with a width of eight words per CP and a store path with a width of four words per CP. However, the two paths cannot operate simultaneously. Chaining of two vector operations is allowed on the machine.

3. Test problems: basic operations

A set of basic vector operations was used in this study, see [2]. A list of these operations is given in Table 1, where a_i , b_i , c_i , d_i , and e_i are vectors and α , β , γ , and δ are scalars. The number of floating point operations, Flop, and vector memory references, Mem, are included in the table. These basic operations were chosen so as to encompass a different number of floating point and memory access operations using a few scalar and vector operands. A straightforward double loop was used, in which the outer loop ran to 100000 and the inner loop ran to $S \times N$ in increments of S , where N is the vector length and S is the stride. Vector lengths of 64, 128, 256, and 512 and strides of 1, 2, and 4 are considered. These lengths were chosen since the length of the vector registers on the SX-2 is 256.

Table 1. List of basic operations.

Loop No.	Operation	Flop	Mem	Flop/Mem
1	$a_i = \alpha b_i$	1	2	0.50
2	$a_i = b_i c_i$	1	3	0.33
3	$a_i = \alpha (b_i + c_i)$	2	3	0.67
4	$a_i = b_i (c_i + d_i)$	2	4	0.50
5	$a_i = \alpha b_i + \beta c_i$	3	3	1.00
6	$a_i = \alpha b_i + c_i d_i$	3	4	0.75
7	$a_i = b_i c_i + d_i e_i$	3	5	0.60
8	$a_i = \alpha b_i + \beta c_i + \gamma d_i$	5	4	1.25
9	$a_i = \alpha b_i + \beta c_i + \gamma d_i + \delta e_i$	7	5	1.40

Table 2. Processing rate (in MFLOPS) of basic operations on the SX-2.

Loop No.	$N = 64$	$N = 128$	$N = 256$	$N = 512$
$I = 1, N, 1$				
1	136.7	263.1	377.3	377.5
2	117.2	215.4	290.1	290.3
3	213.3	394.9	546.9	545.8
4	203.0	364.6	471.5	471.4
5	310.6	503.8	670.2	670.2
6	273.4	432.8	567.9	568.8
7	244.2	435.1	563.9	563.9
8	367.8	630.9	830.0	830.1
9	449.7	735.5	924.7	924.7
Avg.	257.3	441.8	582.5	582.5
$I = 1, 2 \times N, 2$				
1	135.0	234.3	344.1	344.1
2	108.8	179.2	253.9	253.9
3	199.3	333.2	484.8	484.8
4	180.8	290.1	402.5	401.9
5	299.1	467.1	636.8	636.8
6	251.9	386.0	522.4	522.4
7	217.3	338.5	479.4	479.4
8	370.4	584.4	799.0	799.0
9	439.2	693.8	896.9	896.9
Avg.	244.6	389.6	535.5	535.5
$I = 1, 4 \times N, 4$				
1	135.0	227.2	273.4	273.5
2	100.6	156.0	183.9	183.9
3	185.4	290.1	367.8	367.8
4	159.1	235.8	277.0	277.0
5	278.3	413.4	551.6	551.1
6	223.8	269.1	415.4	415.6
7	187.1	217.6	333.3	333.3
8	370.4	451.6	692.2	692.5
9	419.5	507.3	777.8	777.7
Avg.	228.8	307.6	430.3	430.3

Table 2 contains the processing rates (in MFLOPS) of the nine loops on the SX-2. Also, the average values for the nine loops are given for comparison. The highest achieved rate (924.7 MFLOPS) represents about 70% of the peak performance rate of the machine (1333.3 MFLOPS). The lowest rate (100.6 MFLOPS) represents about 8% of the peak rate. The results show that the performance of

the SX-2 improves as the vector length increases up to 256. Beyond this value, no improvement on the performance of the machine was noticed. If we assume that the machine has achieved its peak rates for these loops at vectors of length 256, then the results for vectors of length 64 and 128 represent an average of 44% and 76% of these peak rates, respectively. The results also show the impact of non-unit stride operations on the performance of the machine. Memory strides of 2 and 4 caused an average performance degradation of about 8% and 22%, respectively.

4. Performance model

The following model is based on estimating the number of effective floating point and memory access operations in a code [2]. This model is used to determine the impact of certain architectural features on the performance of the machine. Features such as multiple functional units, overlapping of CPU and memory operations, multiple memory links, and chaining are considered here. The processing rate of a code can be modeled by

$$f_p = \frac{N_f}{N_{eo} CP},$$

where f_p is the estimated processing rate, N_f is the number of floating point operations in the code, N_{eo} is the total number of effective floating point and memory access operations in that code, and CP is the clock period.

Table 3 contains the number of actual operations, number of effective operations, estimated processing rate, estimated processing rate for specific values, and measured processing rate for the basic operations on the SX-2. The results for vectors of length 256 is considered here since this length matches the length of the vector registers. As shown in Table 3, the number of the effective operations is less

than the number of the actual operations because of the overlapping of CPU and memory operations and chaining. This overlapping helped in hiding the floating point operations behind the memory references for the first four loops where the memory operations are the dominant cost. This overlapping and chaining helped to reduce the cost of the other loops.

Table 3. The number of actual operations, number of effective operations, estimated processing rate, estimated processing rate for $\hat{f} = \hat{m} = f/4$, $\bar{m} = f/8$, $c = f/10$ (\hat{f} , \hat{m} , \bar{m} , and c are floating point, first memory reference, second memory reference, and chaining operations), and measured processing rate for the basic operations on the SX-2 for vector length 256 and stride 1.

Loop No.	Actual ops.	Effective ops.	Estimated proc. rate (f_p)	f_p ($\hat{m} = \hat{f} = f/4$, $\bar{m} = f/8, c = f/10$)	Meas. rate
1	$f + 2m$	$2\hat{m}$	$f/(2\hat{m})CP$	333.3	377.3
2	$f + 3m$	$2\hat{m} + \bar{m}$	$f/(2\hat{m} + \bar{m})CP$	266.7	290.1
3	$2f + 3m$	$2\hat{m} + \bar{m}$	$2f/(2\hat{m} + \bar{m})CP$	533.3	546.9
4	$2f + 4m$	$2\hat{m} + 2\bar{m}$	$2f/(2\hat{m} + 2\bar{m})CP$	444.4	471.5
5	$3f + 3m$	$\hat{m} + \hat{f} + 3c$	$3f/(\hat{m} + \hat{f} + 3c)CP$	625.0	670.2
6	$3f + 4m$	$2\hat{m} + \hat{f} + 3c$	$3f/(2\hat{m} + \hat{f} + 3c)CP$	526.3	567.9
7	$3f + 5m$	$2\hat{m} + \hat{f} + 3c$	$3f/(2\hat{m} + \hat{f} + 3c)CP$	526.3	563.9
8	$5f + 4m$	$\hat{m} + 2\hat{f} + 3c$	$5f/(\hat{m} + 2\hat{f} + 3c)CP$	793.7	830.0
9	$7f + 5m$	$\hat{m} + 3\hat{f} + 3c$	$7f/(\hat{m} + 3\hat{f} + 3c)CP$	848.5	924.7

The SX-2 has four sets of functional units, an eight word load path, and a four word store path. In order to estimate the cost of a vector operation, several factors should be taken into consideration. The vector startup time for memory references is 36 CPs. This means that a memory fetch of N elements, for $N \leq 256$, takes $36 + (N/8)$ CPs to complete and a memory store takes $36 + (N/4)$ CPs. However, two consecutive fetch operations can overlap the startup time, and two fetches cost only $36 + 2 + 2(N/8)$ CPs, where the instruction issue time is 2 CPs. Also, fetch operations can overlap with store operations. The vector startup time for addition is 9 CPs while it is 13 CPs for multiplication. However, as for memory

references, two consecutive floating point operations can overlap the startup time, and two additions cost only $9 + 2 + 2(N/4)$ CPs. Based on these values, it is estimated that a first memory reference, \hat{m} , of 256 elements costs roughly as much as a floating point operation, \hat{f} . This cost is about one fourth of an actual floating point operation, f , since there are four sets of functional units. A second memory reference, \bar{m} , costs about half of that, since there is an eight word load path. The cost of a chaining operation, c , depends on the units that are involved in that operation. It is estimated that chaining costs about 10% of an actual operation. These estimated values are used in Table 3 to estimate the processing rate for every loop. Based on these assumptions, the estimated rates are within 12% of the measured values.

5. Comparison

The basic operations were also run on the Cray-2 and Cray Y-MP [2]. Figures 1 through 4 show a comparison of the performance of the basic operations on the SX-2 and one processor of the Cray-2 and Cray Y-MP as a function of the ratio of the number of floating point operations to memory references (Flop/Mem) for four vector lengths. Only the results for stride 1 operations are considered here. These figures show the dependency of the performance of the SX-2 and Cray-2 on the Flop/Mem ratio. The Cray Y-MP is less dependent on this ratio, and reasonable performance can be achieved even for a low ratio. This indicates that the memory access and processor speed are more balanced on the Cray Y-MP than on the SX-2 and Cray-2. Although two words can be fetched and one word can be stored for each set of the functional units of the SX-2, the inability to fetch and store concurrently and the large memory latency (36 CPs as opposed to 19 CPs on the Cray

Y-MP) are the main reasons behind this imbalance in the performance of the machine.

The Flop/Mem ratio gives only a rough estimate of the performance of an operation on a certain architecture. The performance of loop 3 (with a Flop/Mem ratio of 0.67) is a clear example of the impact of other factors; see Figures 1 to 4. Among these factors are chaining, multiplicity of functional units, and so on; see section 4.

Table 4 compares the performance of the SX-2 with one processor of the Cray-2 and Cray Y-MP using the basic operations. As shown in Figures 1 through 4 and Table 4, The SX-2 outperformed the Cray-2 and Cray Y-MP in every case. It is faster than the Cray Y-MP by up to 86% for short vectors and by a factor of up to 3.82 for long vectors. However, the SX-2 needs longer vectors to achieve a good percentage of its peak performance than the Crays. This is because the length of the vector registers on the Crays is 64 while it is 256 for the SX-2.

Table 4. Performance comparison using the basic operations

Vector length	SX-2/Cray-2			SX-2/Cray Y-MP		
	Low	Average	High	Low	Average	High
64	2.62	2.98	3.28	1.01	1.37	1.86
128	4.40	4.85	5.62	1.86	2.36	3.04
256	4.84	5.96	7.71	2.25	2.96	3.82
512	4.41	5.16	6.10	2.16	2.92	3.82

Memory strides of 2 and 4 do not cause a serious problem on the SX-2 and Cray Y-MP while it is a problem on the Cray-2 [2]. A memory stride of 2 resulted in an average performance degradation of 8% on the SX-2 and 7% on the Cray Y-MP while it is 55% on the Cray-2. Also, a memory stride of 4 resulted in an average performance degradation of 22% on the SX-2 and 36% on the Cray Y-MP

while it is 72% on the Cray-2.

6. More tests: numerical algorithms

In order to get a better feel for the performance of the machine, two numerical algorithms were also employed. The first algorithm is a four color cell relaxation scheme for the solution of the Cauchy-Riemann equations. The second algorithm is an ADI scheme for the solution of the diffusion equation. These algorithms were also run on the Cray-2 and Cray Y-MP [2]. Table 4 contains the processing rate (in MFLOPS) and performance ratio of the two algorithms on the SX-2 and one processor of the Cray-2 and Cray Y-MP for three problems of sizes $N \times N$ grid points, where N is 64, 128, and 256. The inner loops of each code were fully vectorized on these machines.

Table 5. Performance of numerical algorithms.

Domain size	Performance rate (MFLOPS)			Performance ratio	
	SX-2	Cray-2	Cray Y-MP	SX-2/Cray-2	SX-2/Cray Y-MP
Cauchy-Riemann Eqs.					
64 × 64	182.9	109.6	177.6	1.67	1.03
128 × 128	310.2	112.9	190.1	2.75	1.63
256 × 256	454.7	115.2	190.6	3.95	2.39
Diffusion Eq.					
64 × 64	205.2	85.6	130.8	2.40	1.57
128 × 128	284.7	88.9	135.2	3.20	2.11
256 × 256	335.7	95.2	136.5	3.53	2.46

The two algorithms achieved 14% to 34% of the peak performance rate of the SX-2. The algorithm for solving the Cauchy-Riemann equations has many stride 2 memory references and vectors of length $N/2$, and has a Flop/Mem ratio of about 2. The ADI scheme uses the Gaussian elimination algorithm for solving two sets of tridiagonal systems, and has a Flop/Mem ratio of about 1. The SX-2 outperformed

the Crays for the three domain sizes of both algorithms. It outperformed the Cray-2 by a factor of up to 4 and the Cray Y-MP by a factor of up to 2.5. Again, the vector length plays an important factor here. These algorithms achieved reasonably high performance on these machines because they are well vectorized and have a high Flop/Mem ratio.

7. The NEC SX-X

The NEC SX-X is an evolutionary step from the SX-2 supercomputer with up to four arithmetic processors and a clock period of 2.9 nsec [3]. The top model, SX-X44, has four arithmetic processors of four pipeline sets each. Each set has two adders and two multipliers which can also be used for other operations. Each arithmetic processor has 72 vector registers of 256 elements each. The number of memory paths to the vector registers has been increased to four load paths and two store paths per processor. Each of the six paths has a width of four words per CP. Also, unlike the SX-2, these paths can operate simultaneously. The machine has also a data control processor, up to four I/O processor, a main memory, and an extended memory. The data control processor is to control I/O operations while each I/O processor can sustain a transfer rate of 250 Mbytes/sec with up to 64 channels. The main memory has up to 256 Mwords (64-bit) organized in 1024 banks. It uses 256 Kbit SRAM chips with a 20 nsec access time. The extended memory has a maximum capacity of 2 Gwords with a peak transfer rate of 2.75 Gbytes/s. The machine runs the SUPER-UX operating system which is based on UNIX System V. The vectorizing compiler is Fortran77/SX. The machine is scheduled to be available in North America during the third quarter of 1990.

In order to estimate the performance of the SX-X relative to the SX-2, certain factors should be taken into consideration. The clock period difference gives a speedup of 2.07 in favor of the SX-X. The four processors of the SX-X44 gives an additional factor of four for the whole machine. Moreover, the replacement of the logical and shift units of the SX-2 by more general add and multiply units will give a speedup of up to two for certain codes. Overall, the SX-X44 is theoretically faster than the SX-2 by 8.3 to 16.6 times. Probably, the balance between the memory access and processor speed will improve for the SX-X. Although the number of add and multiply units has been doubled, the number of memory paths has been essentially doubled. The ability to use the load and store paths simultaneously will help in utilizing the memory more efficiently. The overall improvement in this balance will depend on the type of applications and the ability of the compiler to use the additional units efficiently. Another important factor is the impact of the vector length on the performance of the machine. It is expected that the SX-X will have comparable startup times for memory and floating point operations to the SX-2. This could mean that the SX-X, like the SX-2, will require relatively long vectors to achieve reasonable performance.

The following is a rough estimate of the performance of the machine. The one processor SX-X (SX-X14) will probably be three times faster than the SX-2. Based on the results of this study and previous ones (see [1], [4], [6]), the SX-2 can be considered as twice as fast as one processor of the Cray Y-MP. This makes the SX-X about six times faster than the Cray Y-MP per processor. So, roughly, the SX-X44 will be three times faster than the eight processor Cray Y-MP, even though the former is theoretically 8.7 times faster than the latter.

8. Conclusions

A set of basic operations was used to analyze the performance of the NEC SX-2. The results for four vector lengths, ranging from 64 to 512, and using memory strides of 1, 2, and 4 were analyzed and compared with early results on one processor of the Cray-2 and Cray Y-MP. For unit stride operations, the performance of the SX-2 ranged between 117.2 and 924.7 MFLOPS. This wide performance range was attributed to the vector length and the ratio of processor speed to memory access. It was found that the SX-2 requires longer vectors to achieve a good percentage of its peak performance than the Crays. This is because the length of the vector registers on the SX-2 is 256 while it is 64 for the Crays. Also, the Flop/Mem ratio has more impact on the performance of the SX-2 and Cray-2 than on the Cray Y-MP. A performance model was developed to determine the impact of certain architectural features on the performance of the machine. The model was used to reproduce the measured results with an error of at most 12%. Two numerical algorithms were also implemented to get better feel for the performance of the machine. The results for the basic operations and the numerical algorithms showed that the SX-2 outperformed a single processor of the Cray-2 and Cray Y-MP in every case. It outperformed the Cray Y-MP by up to 86% for short vectors (length of 64) and by 2 to 4 times for long vectors (length of 256). Also, it outperformed the Cray-2 by bigger factors. In addition, memory strides of 2 and 4 do not cause a major performance degradation on the SX-2.

The focus of this study was the vector performance of the machine and the architectural features that have the most influence on its performance. The scalar performance of the machine was not included in this study. Neither was the

impact of the size of the main memory and the speed of the extended memory and I/O devices. More experiments may be needed to provide a better understanding of the different factors influence the performance of the machine.

Acknowledgements

I would like to thank Richard Edwards of HNSX Supercomputers Inc. for his assistance in running the codes and clarifying certain points about the machines.

References

- [1] Bailey, D. H. and Miya, E., "NAS Kernel Benchmark Results," Supercomputer Applications Newsletter, Vol. 3, No.1, pp. 1-4, Feb. 1989.
- [2] Fatoohi, R. A., "Vector performance analysis of three supercomputers: Cray-2, Cray Y-MP, and ETA10-Q," in Proc. Supercomputing '89, Reno, Nevada, Nov. 1989, to appear.
- [3] HNSX Supercomputers Inc., System overview SX-X series, June 1989.
- [4] Luback, O., Moore, J., and Mendez, R., "The performance of the NEC SX-2 and Cray X-MP supercomputers," Los Alamos Tech. Rep. LAUR-87-227, 1987.
- [5] Luback, O., Moore, J., and Mendez, R., "A benchmark comparison of three supercomputers: Fujitsu VP-200, Hitachi S810/20, and Cray X-MP/2," Computer, Vol. 18, No. 12, pp. 10-24, Dec. 1985.
- [6] Van Kats, J. M., Liurba, R., and Van Der Steen, A. J., "A close look at the first generation of Japanese supercomputers," ACCU Tech. Rep. TR-22, Utrecht, Nov. 1986.

- [7] Watanabe, T., "Architecture and performance of NEC supercomputer SX system," *Parallel Computing*, Vol. 5, pp. 247-255, 1987.

FIG. 1. VECTOR LENGTH: 64

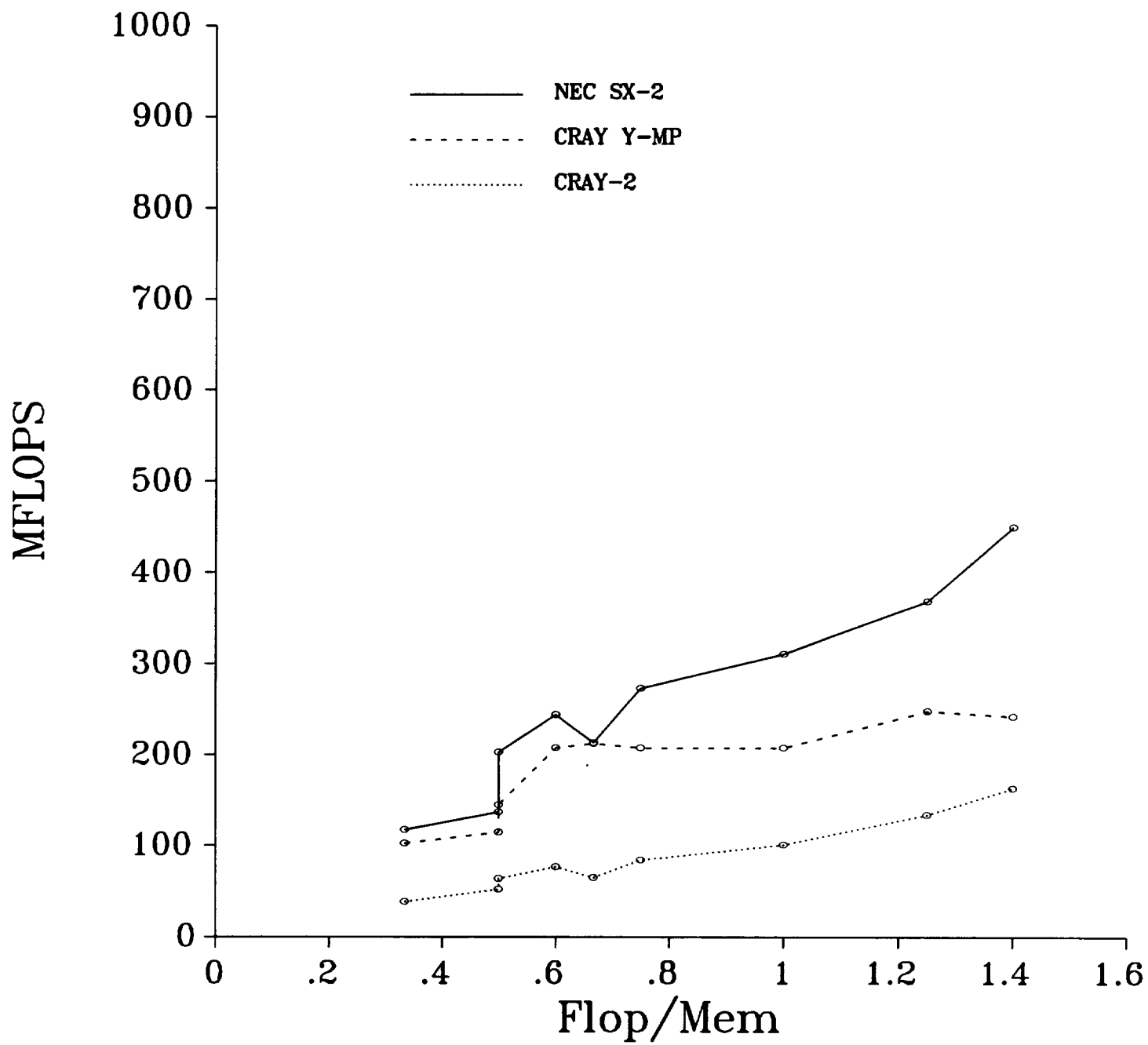


FIG. 2. VECTOR LENGTH: 128

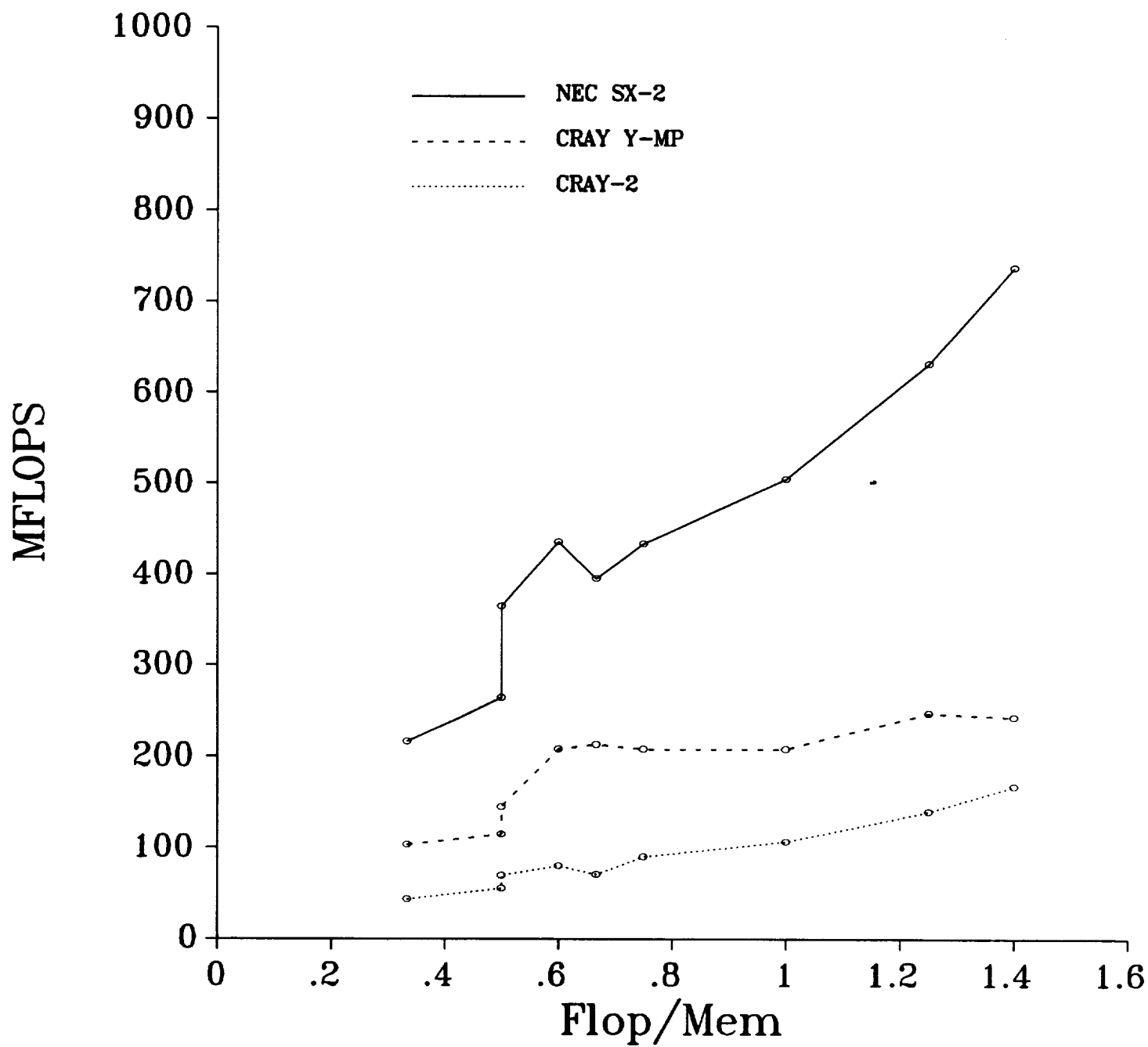


FIG. 3. VECTOR LENGTH: 256

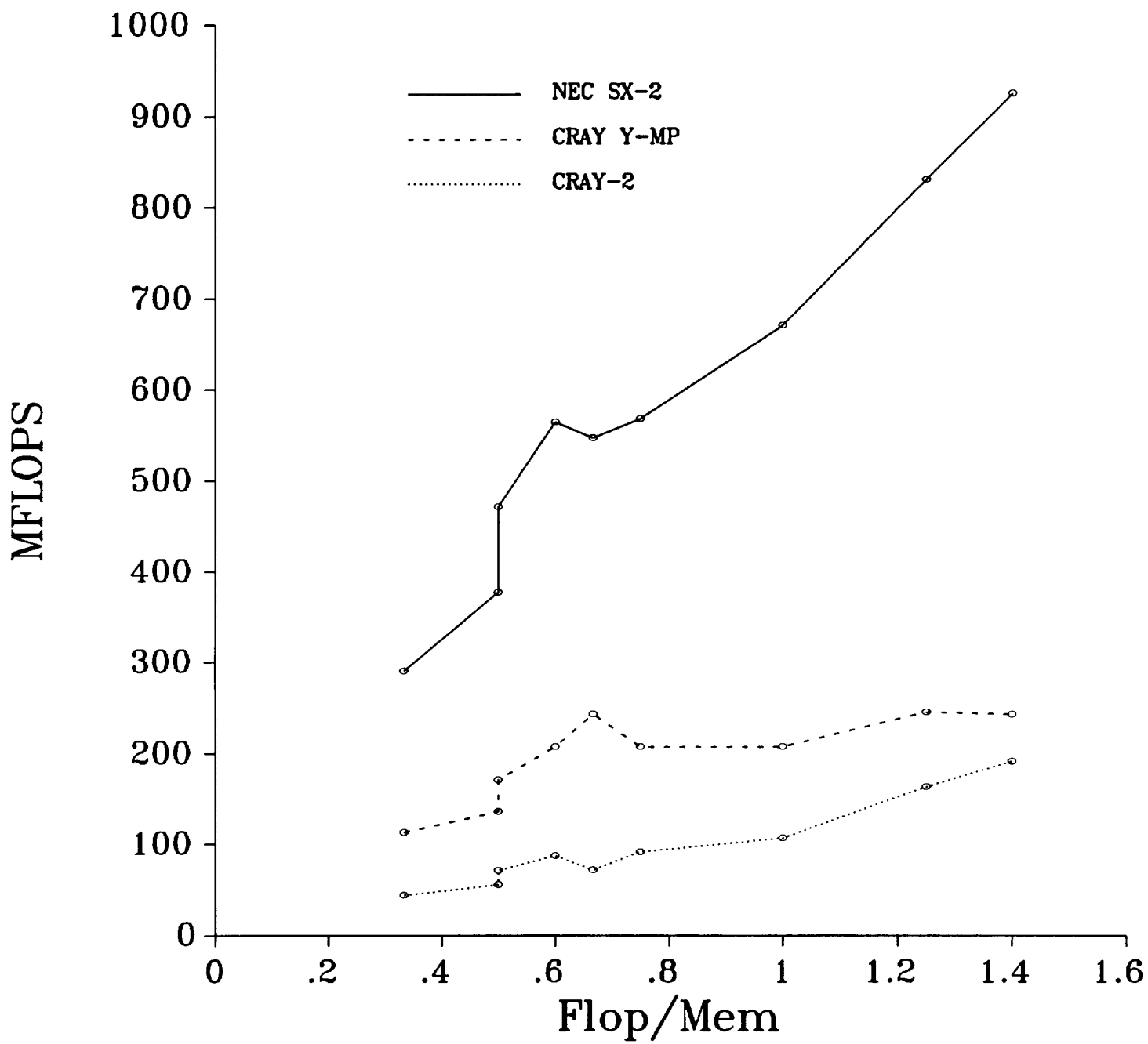


FIG. 4. VECTOR LENGTH: 512

